

A framework-based approach to building private trading exchanges

by S. Kumaran
Y. Huang
J.-Y. Chung

The private trading exchange (PTX) is emerging as the cornerstone of business-to-business e-commerce. At its core, a private trading exchange involves interenterprise integration and collaboration. Our main objective in this paper is to introduce a versatile application platform for building private trading exchanges. This platform supports the management and execution of multienterprise business processes and collaborations in the context of these processes. We review the business requirements of PTX-based systems and the technical challenges in developing such systems. We then examine and categorize the main interaction patterns between the private trading exchange and its participants. These patterns lead to the design of a component framework for process brokering and content aggregation. This framework is the foundation of our application platform for building PTX solutions. We describe the platform and discuss how to use it to implement a PTX solution.

Private trading exchanges (PTXs) are emerging as the cornerstones of business-to-business e-commerce. AMR Research describes the PTX as “an application platform on which a company builds its trading interface to both suppliers and customers via Internet.” They predict the PTX market to garner \$5.7 trillion in commerce transacted via the Internet by 2004, easily surpassing public and consortium exchange markets.¹

How do private trading exchanges differ from other application platforms? What are the functional and nonfunctional requirements of private trading exchanges? What are the technical challenges in realizing such a platform? How can we build a versatile application platform for private trading exchange solutions that addresses these challenges? These are some of the questions we try to answer in this paper.

The heart of a private trading exchange solution is interenterprise integration and collaboration. An application platform for private trading exchange solutions should provide support for managing and executing multienterprise business processes. It should help the employees and trading partners of a company to collaborate, in the context of these business processes. Effective business process execution requires enterprise application integration as well. The platform should support “best-of-breed” application integration, since it is most likely that the diverse applications of an enterprise come from multiple independent software vendors (ISVs). Other requirements include an integrated user interface, common security mechanisms, and solution management and monitoring.

A key nonfunctional requirement is the ability of the platform to adapt to rapidly changing business conditions: new marketplaces and exchanges need to be

©Copyright 2002 by International Business Machines Corporation. Copying in printed form for private use is permitted without payment of royalty provided that (1) each reproduction is done without alteration and (2) the *Journal* reference and IBM copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free without further permission by computer-based and other information-service systems. Permission to *republish* any other portion of this paper must be obtained from the Editor.

integrated; new protocols and messaging standards must be supported; and new trading partners and service providers should be discovered dynamically. Other nonfunctional requirements include performance and scalability, reliability and availability, and extensibility.

We describe the design and implementation of an application platform that meets these requirements. Based on the requirements and use cases, we have identified the fundamental design patterns that are applicable in building a PTX solution. We used these patterns to develop a component framework that serves as the foundation of the platform. Here we focus primarily on two key modules in the platform, the business flow manager that orchestrates multi-enterprise business processes and the interaction manager that manages client interactions with the exchange. We discuss the current realization of the platform, which uses distributed object technology and off-the-shelf middleware.

The paper is organized as follows. In the next section, we discuss the business requirements of business-to-business e-commerce solutions based on the PTX concept and identify the platform requirements. In the following section, we position our work in context by surveying related work in this area. We then relate the requirements to a set of patterns that characterize and categorize the interactions between a company's private trading exchange and its trading partners. This leads to a discussion of the process broker framework and its application in realizing the platform. We describe an implementation of the platform based on WebSphere* Business Integrator.² We then describe an innovative solution assembly methodology for building a PTX solution using this platform. At the heart of this methodology is a PTX solution template, which provides a skeleton to build customized PTX instances, allowing rapid development and deployment. We use a sample PTX solution to present a concrete example of the customized platform. We then revisit the platform requirements and summarize our experience in using the platform with customers. We conclude with future directions of work in this area.

Requirement and challenges

The PTX is a trade exchange hosted by a single company to facilitate collaborative e-commerce with its trading partners. Concerned with the generic nature of public e-marketplaces, a company might develop and host a private exchange in order to gain control

over who may participate, in what manner, how they may be connected, what contents should be presented, and to whom, among many other issues.³⁻⁵ The PTX must support the complex relationships the company may have with its trading partners, ranging from tightly coupled partners in the supply chain processes, to loosely related "spot" buyers and suppliers from e-commerce portals. The ultimate goal might be to improve supply chain efficiencies and responsiveness through improved process visibility and collaboration, advanced integration platforms, and customization capabilities.

At an early stage of a company's e-commerce efforts, external connectivity is the main concern. Early efforts usually result in establishing message-centric and transaction-oriented links with trading partners. Clearly, this is not enough to support the richer interenterprise process collaboration that might be required. For process collaboration, the participants, at minimum, should have the ability to monitor the states of the transactions and manage their transitions. There are basic requirements that differentiate a private exchange from other forms of trade exchanges.

Direct access, control, and connectivity: A PTX should allow the hosting firm to deal with its customers or suppliers more directly and securely, and to control which partners see what data and perform which functions. The PTX platform should support the different access rights of its participants, mirroring those of the trading relationships with the hosting firm. In addition, the PTX platform should permit the coexistence of multiple connection methods, reflecting the existing or emerging connectivity patterns preferred by the host firm's trading partners. Many of the public exchanges, in contrast, promote the use of standard communication protocols and transport mechanisms specific to their particular industries.

Sensitive data sharing with select partners: A PTX should be able to reveal sensitive price and inventory information to chosen customers and suppliers. For instance, component consumption information may be shared with certain suppliers as part of vendor-managed inventory agreements.

Enriched buy and sell capabilities: A PTX should also allow the firm to sell products based on features other than price, by providing additional information to potential buyers. For instance, products bundled with additional service agreements may increase the to-

tal profit margins. This type of bundling can be more effective in a private exchange setting.

Business agreement-based collaboration: A PTX should provide real-time connectivity and collaboration with trading partners, reflecting and supporting supply-chain agreements.

Customizable services and processes: A PTX should allow a high degree of customer collaboration and provide the ability to customize services and technology to a firm's specific supply chain needs and trading relationships.

Event- and trigger-driven processing: A PTX should pass efficiencies along the supply chain through further analysis of sensitive data. For instance, it can flag problems involved in a specific supply pipeline, such as too many components ordered, allowing faster and automated resolution. Private exchanges should be able to handle a richer set of events and triggers than public exchanges, because they have better access to the process and organizational details of the host companies' portion of the supply chains.

These requirements entail the following platform characteristics (PCs):

- Process collaboration (PC1): A PTX should allow the employees and trading partners of a company to collaborate in the context of various business processes.
- Application integration (PC2): Effective business process execution requires enterprise application integration as well. The platform should support application integration, because it is likely that the diverse applications of the firm come from multiple ISVs.
- Integrated user interface (PC3): The integrated user interface with role-based rendering is another basic platform characteristic of the PTX platform. This is to ensure the uniformity of the user interface and its integration with the PTX server components.
- Common and customizable security mechanisms (PC4): The security mechanisms should be customizable, supporting a large variety of collaborative relationships while maintaining commonality in terms of security component administration.
- Solution management and monitoring (PC5): The PTX administration component must meet business control requirements and handle errors and exceptions during operations.

Table 1 shows how the PTX requirements are addressed by the platform design characteristics. In addition to the functional requirements, there are several nonfunctional requirements. These include the ability of the platform to adapt to rapidly changing business conditions (e.g., the integration with other exchanges, support for new protocols and messaging standards, and dynamic discovery of trading partners and service providers), performance and scalability, reliability and availability, and extensibility.

Related work. Even though the PTX is a relatively new phenomenon, many of the underlying technologies for PTX platforms have been developed and used over many years (some may even date back more than a decade, e.g., electronic data interchange [EDI]). We examine related work in three key supporting technologies of a PTX platform to help clarify the distinctive features of the platform proposed in this paper.

First, we discuss *interenterprise connectivity*. At one point, EDI was the only means by which businesses exchanged structured data, using value-added networks (VANs). The appeal of conducting business over the Internet caused companies to look toward Internet-based interenterprise platforms in order to increase flexibility and reduce (VAN) costs, leading to the rising popularity of Web-based protocols such as HTTP (HyperText Transfer Protocol). However, process integration has not been a major consideration in interenterprise connectivity until quite recently. RosettaNet (Electronics⁶) and CPFR (Collaborative Planning, Forecasting, and Replenishment⁷) are the first two to emphasize the importance of definitive process support in interenterprise collaboration. Today, it is generally acknowledged that a PTX platform should support end-to-end process collaboration through the coordination of inter- and intraenterprise processes.

Second, a PTX platform requires the integration of many applications and systems as part of process collaboration. In the past, integration of disparate applications and systems was typically done through highly customized "hooks" with point-to-point links. This is not desirable from a development and maintainability standpoint. Furthermore, these efforts in many cases assume the passive, or batch, mode of participation for these applications and systems in the overall solution. This assumption does not hold when a PTX solution requires invoking an application function in real time as part of the collaboration. The application integration technology in a

Table 1 Mapping of requirements to the platform characteristics

Requirements	Platform Characteristics
Direct access, control, and connectivity	Through a unified portal (PC3), user registration and management processes (PC1), and role-based access control (PC4)
Sensitive data sharing to select partners	By the role-based control (PC4) on business data object access (PC1, PC2)
Enriched buy and sell capabilities	Through a process (PC1) that spans multiple business units (intraenterprise) and trading partners (interenterprise), and application and data sources (PC2) belonging to these entities
Business agreement-based collaboration	By a collaborative process (PC1) with proper integration with business data and applications (PC2) based on the trading partner profile (PC4) supported by an interaction portal (PC3)
Customizable services and processes	Through a customization portal (PC3) specific to a particular trading partner (PC4) to define desired processes (PC1) coupled with applications and services (PC2)
Event- and trigger-driven processing	By monitoring events relating to solution execution (PC1, PC2, PC5) and triggering proper error/exception handling processes (PC1, PC5) through browser-based reports and other electronic notification methods (PC3) to predefined administrator roles (PC4)

state-of-the-art PTX platform should support event-driven and interactive invocation of application functionality in a process execution context.

Finally, a PTX platform must support complex business processes. Business objects (containing both data and logic) have been used to support the encapsulation of complicated access to legacy systems so as to hide it from the application clients. It relieves the business processes from having to be concerned with the details of rigid legacy systems and legacy architectures. However, they still require the process clients to perform complex maintenance and control tasks, which reduces application maintainability and extensibility. Workflow management addresses some, but not all of the issues. The new generation of PTX platforms needs to support optimal business process management.

These areas have developed independently over the years and accumulated a substantial amount of applicable technology to address individual and sep-

arate concerns. But the framework that brings all these together in a cohesive way to address the total PTX requirements and challenges is lacking. In this paper, we attempt to address this gap by introducing a component framework for integration and collaboration and proposing a new PTX platform based on this framework.

The important role that the PTX platforms play in large companies' overall enterprise computing strategies makes them one of few remaining bright spots in the enterprise software market. It is not surprising that many independent software vendors (ISVs) are gearing up to serve this segment of the software market. We review several ISVs and discuss how they attempt to address the issues.⁸⁻¹⁴

webMethods¹⁵ has two main products: webMethods Enterprise and webMethods B2B Integration Server. It focuses on the XML (Extensible Markup Language)-based B2B (business-to-business) collaboration that is supported by a suite of products acquired or internally developed, ranging from an integration en-

engine and adapters to B2B management tool sets. Recently, it also added workflow management function to its product portfolio through an acquisition. The main challenges of using webMethods to build private exchanges are in the areas of transaction coordination and management, B2B partner management, and new adapter development for enterprise application integration.

Vitria's¹⁶ main product is BusinessWare. Its underlying products build on top of the Communicator, its CORBA^{**}/IIOP^{**}¹⁷ bus. It stresses the importance of business process management in B2B automation and provides tools to support the definition, configuration, and maintenance of business processes. The key challenges in using Vitria to develop private exchanges involve workflow management, portal development, custom coding, and message transformation.

TIBCO¹⁸ is moving from its messaging roots to become a B2B company. TIBCO has a strong reputation in the financial industry due to its affiliation with Reuters. It is a recognized enterprise application integration (EAI) company. Its main products include ActiveEnterprise for internal system and process integration, ActivePortal for extending business processes to employees, customers, and suppliers, and ActiveExchange for connection with other trading partners and e-markets. The primary challenges of using TIBCO to build private exchanges include the integration issues (transformation, transactional and intelligent routing, and legacy access capability) and business process management.

These examples are representative of the ISVs present in this market, but are far from exhaustive. Most of these vendors are strong in one area but weak in others. The goal of this paper is to present a component framework, based on design patterns, that addresses all key issues in building private exchanges. The implementation of the framework could potentially make use of one or more of the ISV products available in the market.

Our work draws from the existing work on design patterns and component frameworks^{19–23} as well as electronic marketplaces.^{24–31} In particular, the concepts introduced by Nayak et al.,²³ Schmid and Lindemann,²⁷ and Gawlick²⁸ have greatly influenced the design of the application platform discussed in this paper. Nayak et al. present the layered architecture of a virtual corporation management system with the capability for creating, operating, and eventually dis-

solving virtual enterprises. Schmid and Lindemann describe the elements of a reference model for electronic markets. Gawlick discusses the infrastructure for Web-based application integration.

Interaction patterns

The PTX and its participants may interact with each other in many different ways. The PTX portal serves as the single entry point for the trading partners to access all the browser-based PTX functions. Users of a PTX system may view business information (e.g., order status, inventory levels, forecast changes, etc.) or conduct more elaborate collaborations for various supply chain processes (e.g., joint product supply forecast, replenishment, and planning as exemplified in CPFR) via the PTX portal. In this section, we capture the common interaction patterns within a PTX and categorize these patterns. These patterns build a foundation for our further discussions on the PTX platform. Figure 1 presents an overview of interaction patterns discussed in this section.

We examine the interactions with the PTX from two dimensions: *functional control* and *process collaboration*. The first considers primarily what a participant can do on the PTX (e.g., information viewing, transaction control, etc.) while the other looks at how the PTX solutions handle the collaboration among the participants and system components through business process management. The identification of these patterns helps to crystallize the requirements and leads us to the framework and platform design.

Functional control. The simplest function provided by a PTX is that of an *information-viewing* portal. This includes the rendering of both the business content and the status of transactions. Figure 2 illustrates this pattern. Beyond information viewing, a PTX usually allows certain forms of *data/file exchange*, e.g., download and upload of data via FTP (File Transfer Protocol), messaging, EDI, or HTTPS (Secure HyperText Transfer Protocol). The data/file exchanges may involve direct server-to-server communications; Figure 3 shows this pattern. In a more advanced form, a PTX permits the initiation and control of transactions, either manually or programmatically. An automated process execution environment involving transaction management will require such a level of control. This pattern is illustrated in Figure 4. Table 2 lists the interaction patterns in the functional control category.

Figure 1 Overall interaction patterns

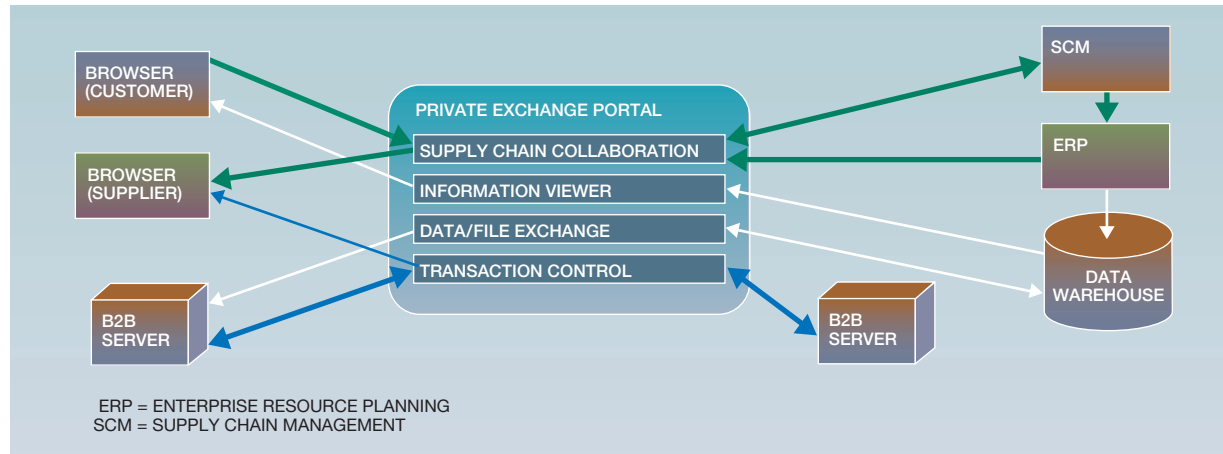
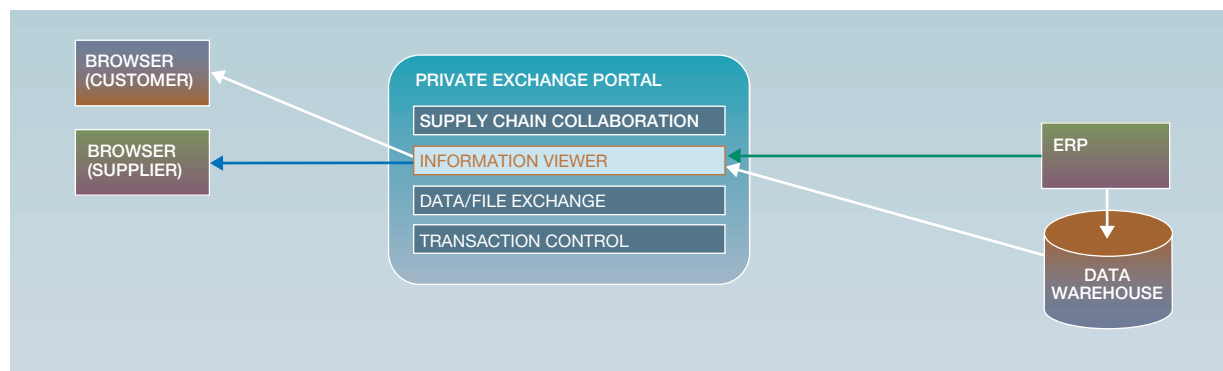


Figure 2 Information viewing interaction pattern



Process collaboration. One of the main features of a PTX is the support of process collaborations, which can take many different forms. A process collaboration (see list in Table 3) can describe the interactions within the enterprise system's components and role players (*intraenterprise*, shown in Figure 5), among the exchange participants (*interenterprise browser*, shown in Figure 6), between the participants' B2B servers and those of the PTX (*interenterprise program*, shown in Figure 7), or between other public or private exchanges and the PTX itself (*collaborative*, Figure 8). An actual PTX typically contains a combination of several such patterns to form end-to-end process collaboration.

Platform description

The patterns discussed in the previous section illustrate the primary interactions among the role players and system components exhibited in a typical PTX solution. These patterns provide the basic design considerations for the development of a component framework that serves as the foundation of the PTX platform. We call this the Process Broker framework, since the two principal functions are (1) brokering of multiple business processes encapsulated in various back-end systems, including workflow engines and business applications, and (2) aggregating content from multiple enterprise information systems, in the business context, and managing its shared ac-

Figure 3 Data/file exchanges interaction pattern

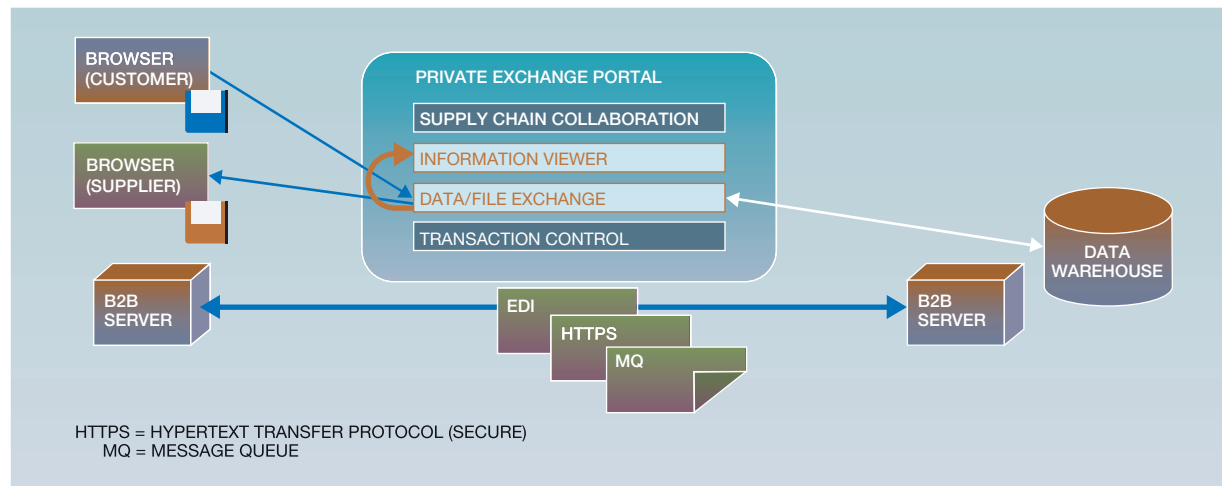
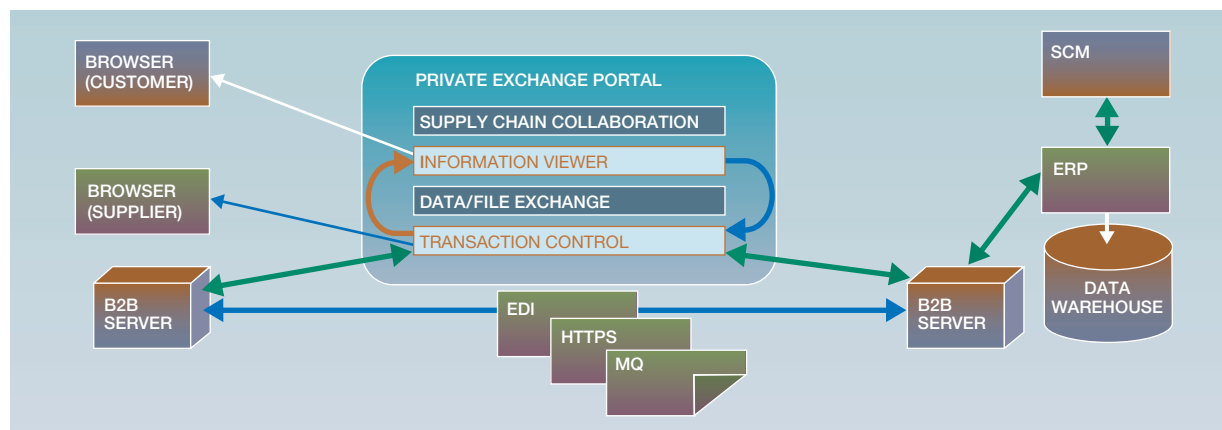


Figure 4 Transaction control interaction pattern



cess based on the roles of participants. These two functions highlight the fundamental business process management capability of our PTX platform.

The goal of the framework is to support rapid development of PTX solutions. It achieves this by creating a brokering layer between the clients of the PTX and the enterprise systems. This layer maps service requests from the clients to appropriate enterprise systems, such as workflow engines, legacy applications, ISV applications, and business objects. It supports browser clients, facilitating human interaction with the system, as well as program clients that rep-

resent business processes running external to the system. Major components of the framework are shown in Figure 9.

We describe the framework using the fundamental design patterns¹⁷ on which it is based. The Process Broker Facade component is designed based on the Façade design pattern. Its purpose is to serve as an entry point for all client requests and to route them to the appropriate brokering components. The brokering components are called adaptive documents (ADocs). ADocs provide domain-specific and state-dependent service brokering by linking the content

Table 2 Interaction patterns—functional control

Types	Characteristics	Typical Examples	Illustration
Information viewing	View general contents View transaction status	Pricing, inventory and point-of-sales, order status	Figure 2
Data/file exchanges	Exchange structured business data through the Web or other May use portal to download/upload or view transactions	Simple: Up/download, FTP, etc. Sophisticated: HTTPS (HyperText Transfer Protocol-Secure), EDI, MQ (Message Queue)	Figure 3
Transaction control	Control in-progress transactions Start, stop, pause, and restart transactions	Transaction exception handling, automatic rollback, manual resent	Figure 4

Table 3 Interaction patterns—process collaboration

Types	Characteristics	Typical Examples	Illustration
Intraenterprise	Between system components and role players within enterprise	Purchase order validation	Figure 5
Interenterprise browser	Between trading partners and the PTX Browser-based	Collaborative planning, forecasting, and replenishment (CPFR), request for quote (RFQ)	Figure 6
Interenterprise program	Between trading partners and the PTX B2B server-based	RosettaNet PIPs, transportation planning and execution	Figure 7
Collaborative	Between public e-marketplace/trading partner intranet and the PTX	Purchases made at a public e-marketplace supported by the fulfillment processes via the PTXs of member companies	Figure 8

aggregated from various data sources to business processes and individuals. They support collaborative business process management through a variety of applications and user interactions, in the context of a business process. For example, a purchase order ADoc supports collaboration between the buyer and supplier organizations in the context of procurement-related processes. ADOcs are based on

the State design pattern,¹⁷ as they demonstrate state-dependent behavior. An ADoc implementation consists of an ADoc entity that captures the state information and an ADoc controller that encapsulates the state-dependent ADoc behavior. The behavior is shared among all instances of an ADoc type. For example, all purchase orders exhibit the same behavior and we need only one controller for the pur-

Figure 5 Intraenterprise interaction pattern

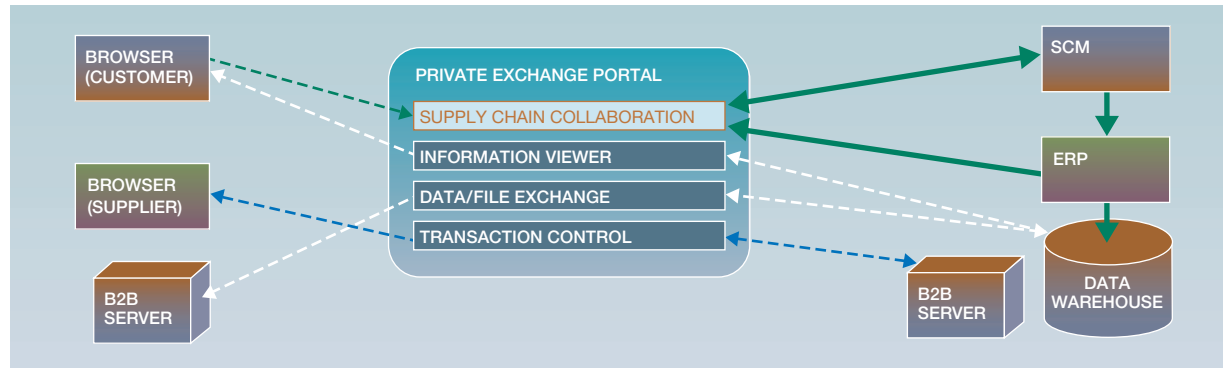
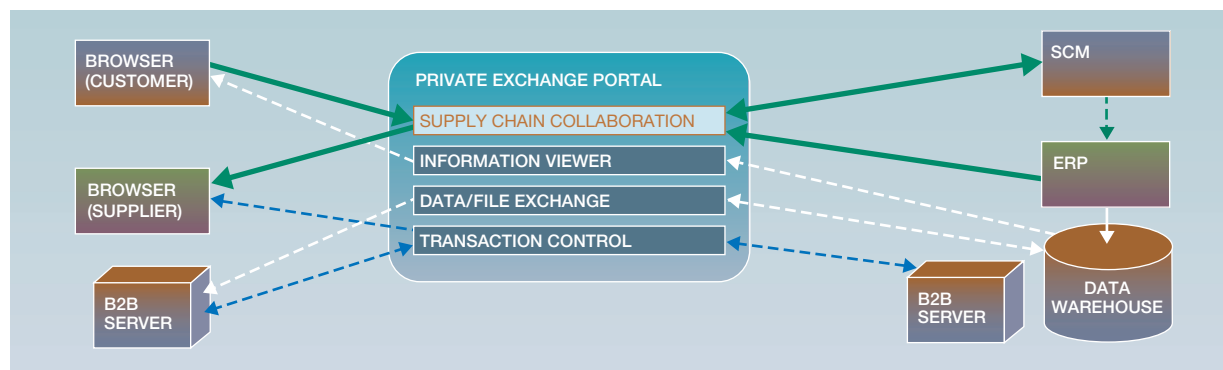


Figure 6 Interenterprise browser interaction pattern



chase order ADoc. An ADoc entity exists for each instance of an ADoc. Using the purchase order example, there is a purchase order ADoc entity for each purchase order in the system.

The framework supports Web transactions by means of ADoc behavior. When a client issues a Web transaction, it reaches some ADoc in the system as a service request. As defined in the behavior of the ADoc, the ADoc controller executes a set of actions determined by the current state of the ADoc and the contents of the service request. These actions are implemented as a transaction. A service request may also result in changing the state of an ADoc. The fulfillment of a service request follows the Command design pattern.²⁰ There are three major participants in the Command design pattern: (1) A Command object declares an interface for executing an operation. (2) A Concrete-Command object defines a

binding between a Receiver object and an action. It implements a specific operation by invoking the corresponding operations on a receiver. (3) A Receiver object knows how to perform the operations associated with a request. Any class may serve as a Receiver class—commonly used receivers include workflow engines, enterprise applications, business objects, and trading partner gateways.

The Process Broker framework is used to realize two key functional modules in the platform as shown in Figure 10: (1) the business flow manager (BFM) module that manages the internal (private) business processes of the enterprise hosting the PTX, and (2) the interaction manager (IM) module that manages the external (public) processes the enterprise exposes to its trading partners. These external processes include both browser and program interfaces. The PTX por-

Figure 7 Interenterprise program interaction pattern

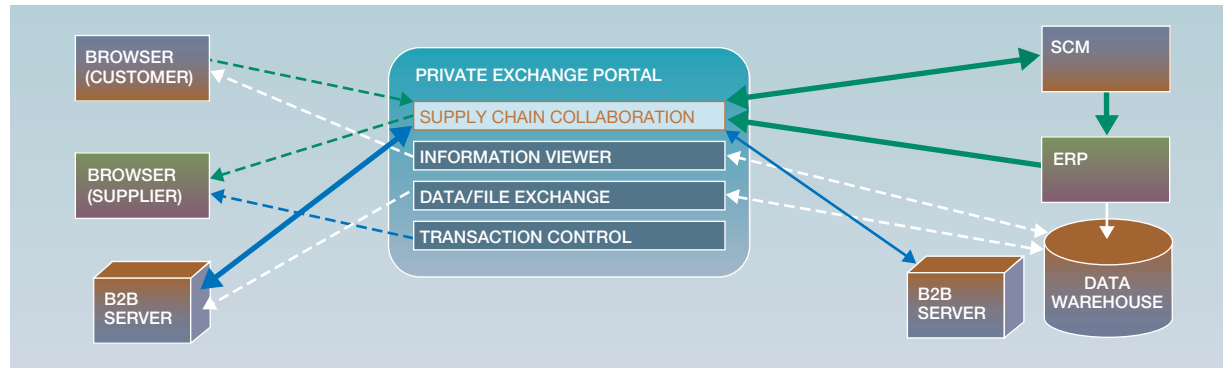
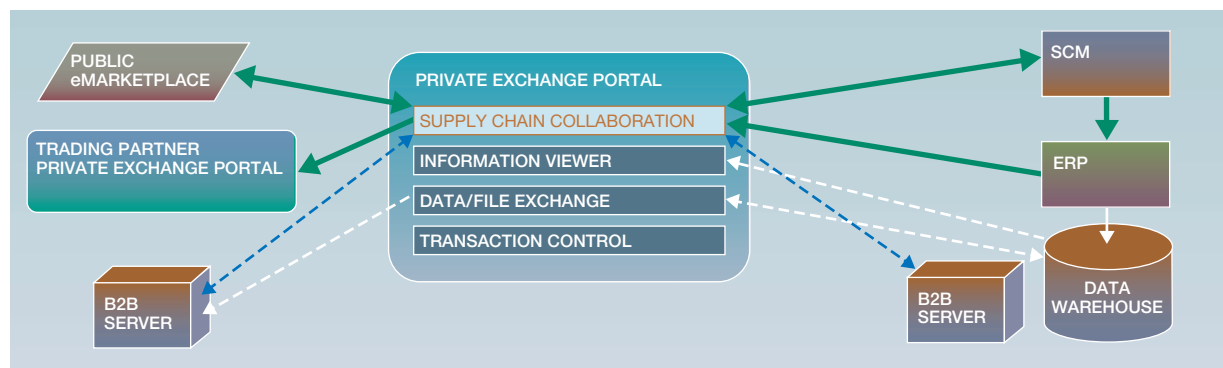


Figure 8 Collaborative interaction pattern



tal discussed earlier is facilitated by the interaction manager component.

These two modules are positioned on top of a base service layer to realize the PTX platform. The base layer consists of the following core services:

- Messaging services, which provide XML-based asynchronous communication support including message brokering, transformation, translation, aggregation, splitting, routing, and publish/subscribe. Java™ Messaging Service (JMS)³² is used as the interface to integrate BFM and IM modules with the messaging services provider.
- Solution management services, which include audit, exception handling, and monitoring of business processes. Java reliability and availability services (JRAS) and Java Management Extensions (JMX)²⁸ are used to define the interface between

the IM and BFM modules and the solution management service provider.

- Authentication and authorization services, which include security, directory support, and role-based access control to the PTX functions. This interface is based on the Java Authentication and Authorization Service.³³

The platform includes a build-time suite of tools called Private Exchange Solution Builder. The solution builder enables creation and customization of PTX solution instances. The high-level component architecture of the platform is shown in Figure 10. “PTX Solution” refers to an actual private exchange solution implemented on the platform, which is discussed in detail later.

The internal processes of a PTX typically involve workflow. The business flow manager module in-

cludes a workflow services component that augments the Process Broker framework. The process broker interacts with the workflow services layer via activity controllers. An activity controller is defined for each node in a workflow graph. The activity controllers are defined exactly the same way as ADoc controllers, using the State and Command design patterns.

A business transaction may need to interact with several concurrent workflow processes, and an ADoc is responsible for the choreography of these processes. An ADoc controller is associated dynamically with activity controllers to effect workflow choreography. For each workflow process involving an ADoc, a corresponding activity controller is associated with the ADoc controller.

The interaction manager uses the process broker framework to implement trading partner protocols and public processes. For example, the RosettaNet PIP** (Partner Information Process) for purchase order processing⁶ is implemented using an ADoc in the interaction manager. Additionally, it provides connection services, session management, and single sign-on (SSO). The ADocs in the interaction manager use the Process Broker Façade object of the business flow manager as a receiver to transmit client transactions into the enterprise.

Platform implementation

The implementation of the private exchange platform is based on IBM WebSphere* Business Integrator (WBI).² WBI is a complete application platform that allows the design, development, and deployment of intra- and interenterprise business processes. In addition to the interaction manager and business flow manager modules discussed earlier, WBI includes the following: solution manager for solution management services, trust and access manager for security and directory services, partner agreement manager for B2B connectivity services, and information delivery manager for messaging services.

The WebSphere Business Integrator programming model is based on the IBM Application Framework for e-business.³⁴ WBI brings together several IBM products to provide a cohesive, prescriptive approach to creating e-business solutions. Table 4 shows the specific IBM products^{35–38} used to implement the PTX platform modules within WBI.

Figure 9 Process broker framework

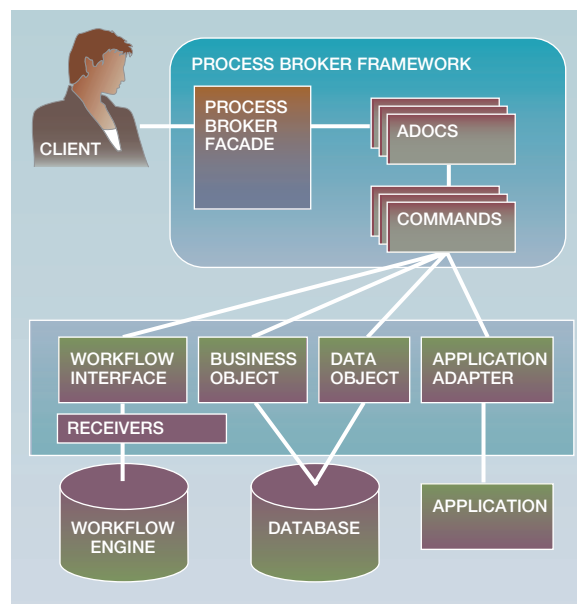
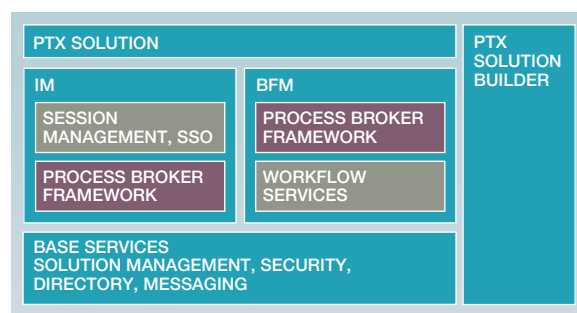


Figure 10 Components of the PTX platform



The process broker framework is implemented as an Enterprise JavaBeans** (EJB**) ³⁹ application on the WebSphere Application Server. Clients communicate with the process broker in one of two ways: (1) synchronously using RMI (Remote Method Invocation) over IIOP or (2) asynchronously via XML messages to a JMS listener in the WebSphere Application Server. Similarly, the process broker communicates with other systems either synchronously using RMI over IIOP or asynchronously via XML messages.

Figure 11 Components of a process template

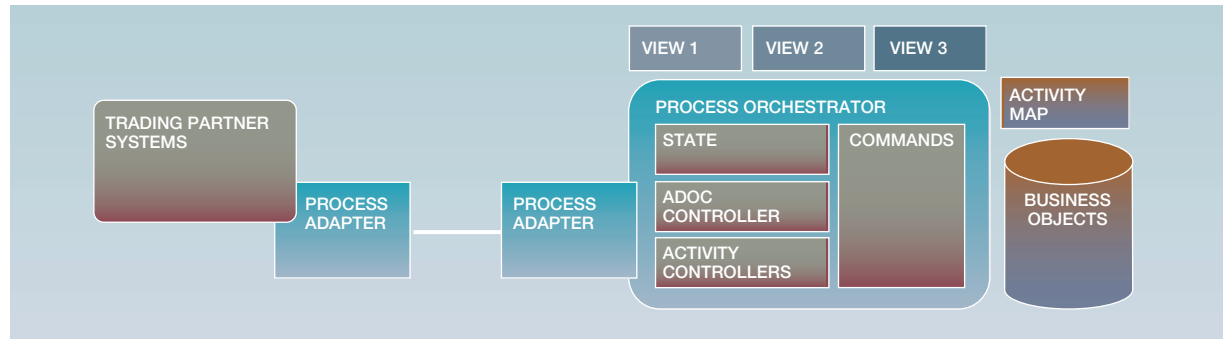


Table 4 IBM products used in the PTX platform implementation

PTX Platform Module	IBM Product
Interaction manager	WebSphere Application Server, WebSphere Portal Server
Business flow manager	WebSphere Application Server, MQSeries*, MQ Workflow
Trust and access manager	Tivoli SecureWay LDAP Server and Policy Director
Information delivery manager	MQSeries Integrator, MQ Adapter Offerings
Solution manager	Tivoli eMarketplace Manager

Solution example

In this section, we describe the implementation of a PTX solution on the platform, derived by customizing a PTX solution template. A solution template consists of a set of predefined process templates, brought together by the process broker. For example, a simplified PTX solution template for the distribution industry may be composed of three process templates: CPFR, RFQ (request for quote), and PO (purchase order)/invoicing.

Figure 11 shows the internal structure of a process template. The components are:

- *Process orchestrator.* This is built on the ADoc concept discussed earlier. A process orchestrator may contain multiple ADocs, which implement the internal business processes of the private exchange and are deployed in the business flow manager module.
- *Activity map.* For each activity, an activity map lists the set of activities to be activated upon its completion. This set may be generated dynamically at run time or statically defined at build time. In some cases, only the activity names are defined at build time and the cardinality is determined at run time. The activity maps define the workflows and are deployed in the workflow engine.
- *Process adapter.* This is a specialized ADoc that handles program-to-program communications between enterprises. Process adapters are deployed in the interaction manager module.
- *Business objects.* These are domain-specific objects that serve as receivers to the commands in the controllers. They include application adapters that connect to legacy applications, as well as new com-

ponents developed to cover “white space” functions in the solution.

- *Views*. Users participate in the business process through ADoc views, which have three components: action, navigation, and content. The *action* corresponds to the dynamic business services that are exposed by an ADoc, depending on the role of the user and the context of the business process, and invoked via the Process Broker Façade interface. The *navigation* can be either process-centric (e.g., the set of actions given the current process state) or document-centric (e.g., the list of ADocs that require some action). The ADoc supports both navigation models, which are accessed via the Process Broker Façade interface. The *content* in the view is marshaled by the execution of the business services in an ADoc, i.e., the business transactions that are brokered by the ADoc controllers to aggregate the content from a multitude of back-end sources, such as applications, business objects, and databases.

Figure 12 shows an actual RFQ template, without the views. Typically, the template will include the views for each role player who is a collaborator in the process. The template has no process adapters, because there are no program-to-program communications across the enterprise boundaries for this process. The artifacts in the template include an ADoc Entity object that maintains the ADoc state, an ADoc controller, activity controllers, activity maps, commands used by the controllers, and receivers. The artifacts shown in gray are provided as part of the PTX platform, while the other artifacts are created as part of the RFQ process template.

The RFQ ADoc, including the RFQ ADoc entity and RFQ ADoc controller, all the commands, and all the activity controllers in Figure 12 map to the Process Orchestrator part of the process template. The “RFQ Processing” and “Quote Create” macro flows map to the Activity Map in the template. These flows are shown in more detail in Figure 13. The receivers (RFQ Business Object [BO], etc.) map to the business objects in the template.

Figure 14 shows the RFQ ADoc controller state machine. The states of the RFQ ADoc, the transitions allowed at each state, the event that triggers each transition, and the commands invoked as part of the transition are shown.

There are five activity controllers in the RFQ Process template. Figure 15 shows the activity control-

ler for the “Evaluate Response” activity. All activities have an “Available” state, a “Claimed” state, and a “Complete” state. Additional states may be added as demanded by the micro workflow associated with an activity.

Experience and evaluation

We have used the platform to build proof-of-concept private exchange solutions for several customers from diverse industry segments, including aerospace, automotive, retail, and electronics. Our experience in building these solutions has underlined the versatility of the platform for the rapid assembly of B2B e-commerce solutions. In this section, we revisit the requirements for building a PTX exchange solution and evaluate the platform against them.

For illustration, we use a sample PTX implementation based on real-world PTX solutions developed for our customers. The solution template is made up of three process templates, as discussed in the previous section. We focus on the RFQ and PO processes here. The RFQ process starts with a buyer, in the company that hosts the PTX, issuing an RFQ request through the PTX portal. The suppliers use the portal to respond to the RFQ, the PTX system aggregates the responses and presents them to the buyer for supplier selection, and the process ends with a PO transaction between the PTX and the supplier system.

Figure 16 shows the detailed sequence diagram for this process. The boxes refer to three types of components: platform components, solution components, and users or partner systems. Supplier and buyer boxes represent the users with the corresponding roles. “Buyer program” represents an information technology system in the buyer enterprise. Gateway and IM components in the figure map to the interaction manager in Figure 10. AM, log server, and BO messaging service components in Figure 16 map to the base services in Figure 10. The BO and SAP** adapter boxes in Figure 16 are parts of the PTX solution. BFM and JAWS** boxes map to the BFM component in Figure 10. IM and BFM boxes in Figure 16 represent parts of the PTX solution, in addition to the platform components.

Now we revisit the requirements introduced earlier and demonstrate how they are addressed by the platform.

Direct access, control, and connectivity. This implementation provides two primary mechanisms for

Figure 12 An RFQ process template

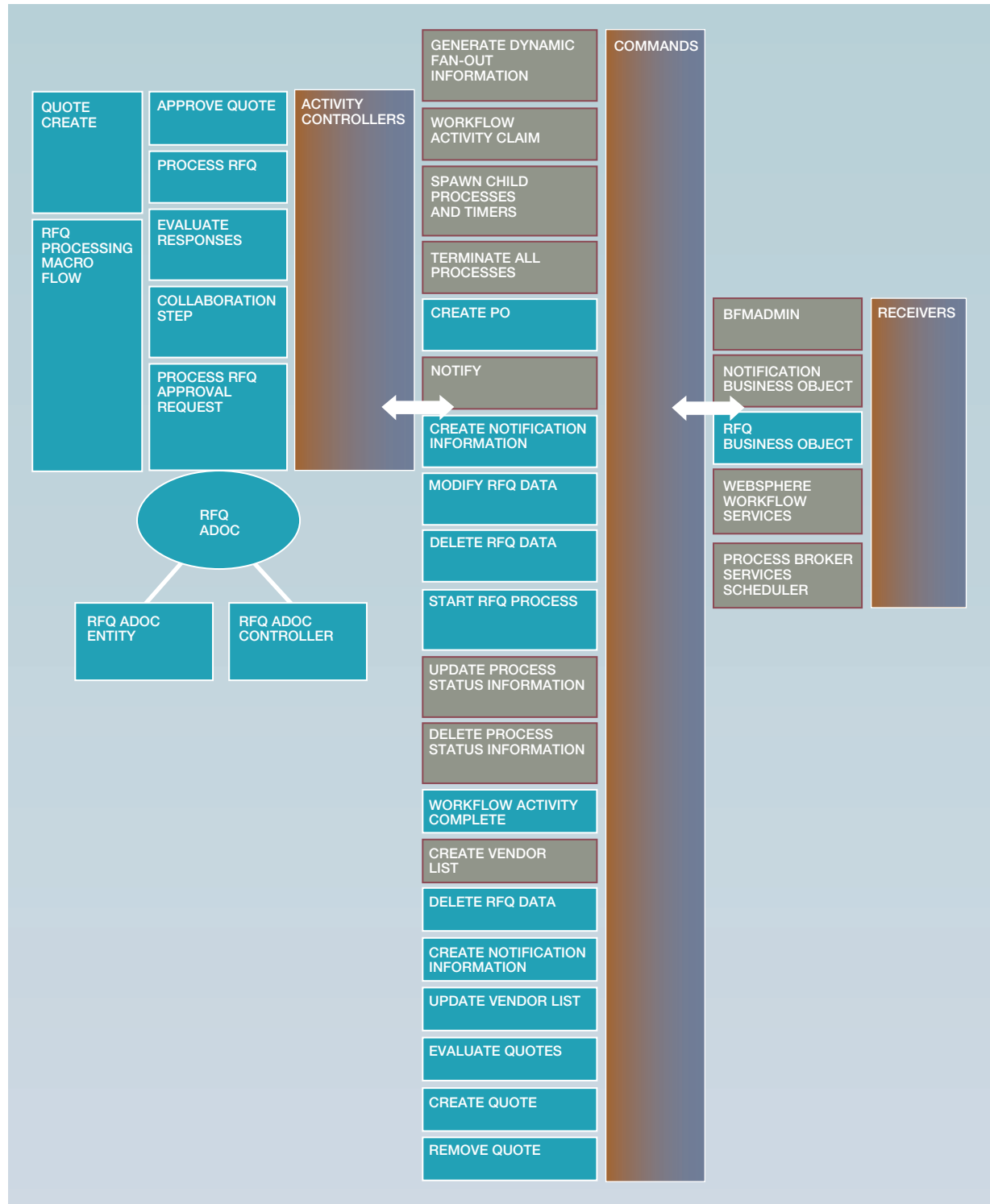
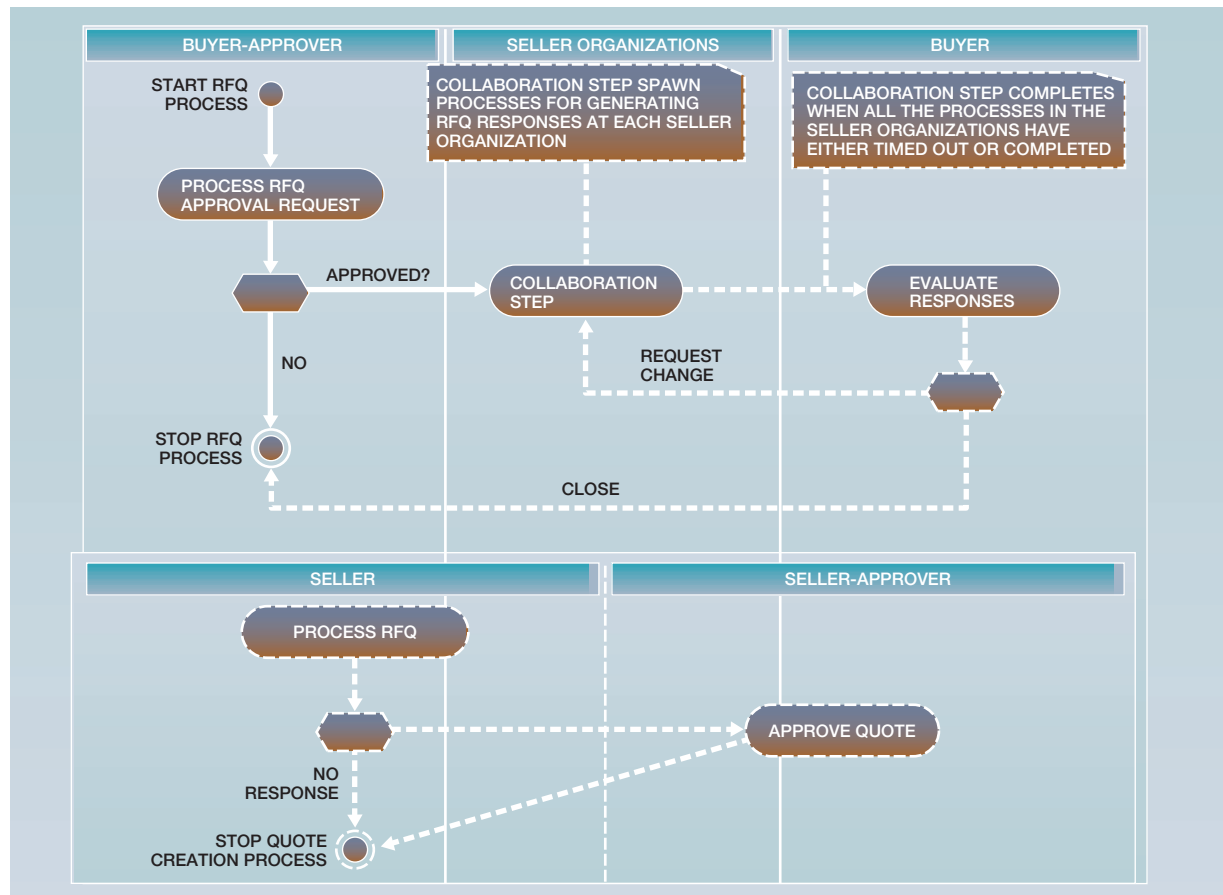


Figure 13 Activity maps for RFQ processing (above) and quote creation (below) in the RFQ process template



trading partners to connect to the PTX: (1) browser-based access via the portal, and (2) programmatic access via a gateway component. In either case, the access manager component provides security service in the platform, by authenticating the user. Additionally, the access manager identifies the user's role (buyer or supplier). The role information is used for access control—the user can access only those system resources authorized for the role. The interaction manager uses this information to render a role-specific view of the portal (supplier portal or buyer portal) and the business flow manager uses it to give user-selective access to system functionality in the management of the entire RFQ process. For the PO transaction, the supplier systems connect with the PTX gateway for program-to-program data exchange. A public process and an associated protocol defined between the PTX and the trading partner manage this

exchange. The ADoc technology is used to implement this public process and the protocol.

Sensitive data sharing with select partners. As just described, the platform has total control over what business data the users can see and which system functions they can invoke. The user security profile (roles and access control list) determines this. In the RFQ example, only the buyer can view the responses from all the suppliers; the competing suppliers can view only their own quotes. The RFQ ADoc realizes this constraint by permitting only the buyer who initiated the RFQ to view the responses from the suppliers.

Enriched buy and sell capabilities. The platform allows the integration of distributed and fragmented data in the context of the RFQ process. For instance,

Figure 14 RFQ ADoc controller state machine

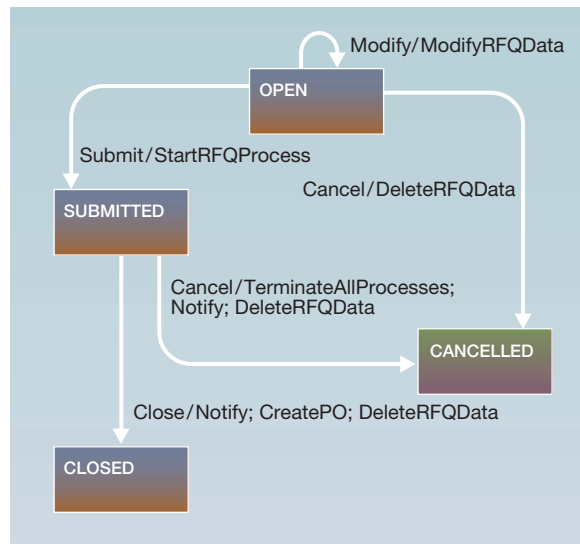
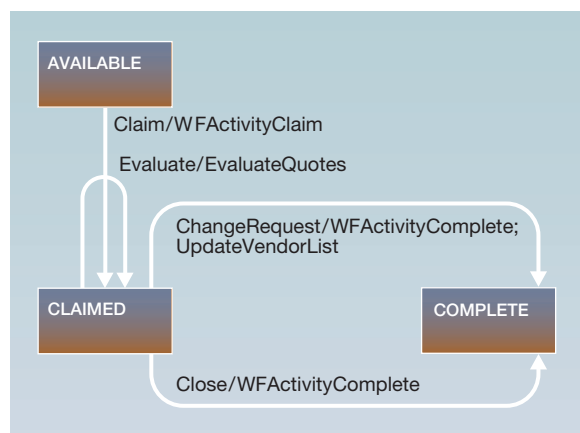


Figure 15 Activity controller for “Evaluate Response” activity



the lead time and the supplier performance history can help the buyer in the quote evaluation and supplier selection process. This is realized through the information integration capability of the activity controller that executes the supplier selection micro workflow (the supplier performance data may be stored in a separate system or database).

Business agreement-based collaboration. The business agreements are externalized and captured in the form of a *trading partner agreement*. The platform uses ADocs to implement these agreements and drive collaborative processes based on them. For example, the platform can support RosettaNet protocol for the PO process with one supplier and a non-standard vendor-specific protocol with another.

Customizable services and processes. The system provides differentiated services and processes by capturing service and process definitions in a trading partner profile, managed by the access manager module. The IM and BFM modules access this information to tailor system behavior to match user preferences. As an example, the RFQ notification methods available to the supplier community may vary based on supplier preference (i.e., some may want e-mail notification while others may prefer a program-to-program RFQ transaction). The trading partner profile includes this type of information.

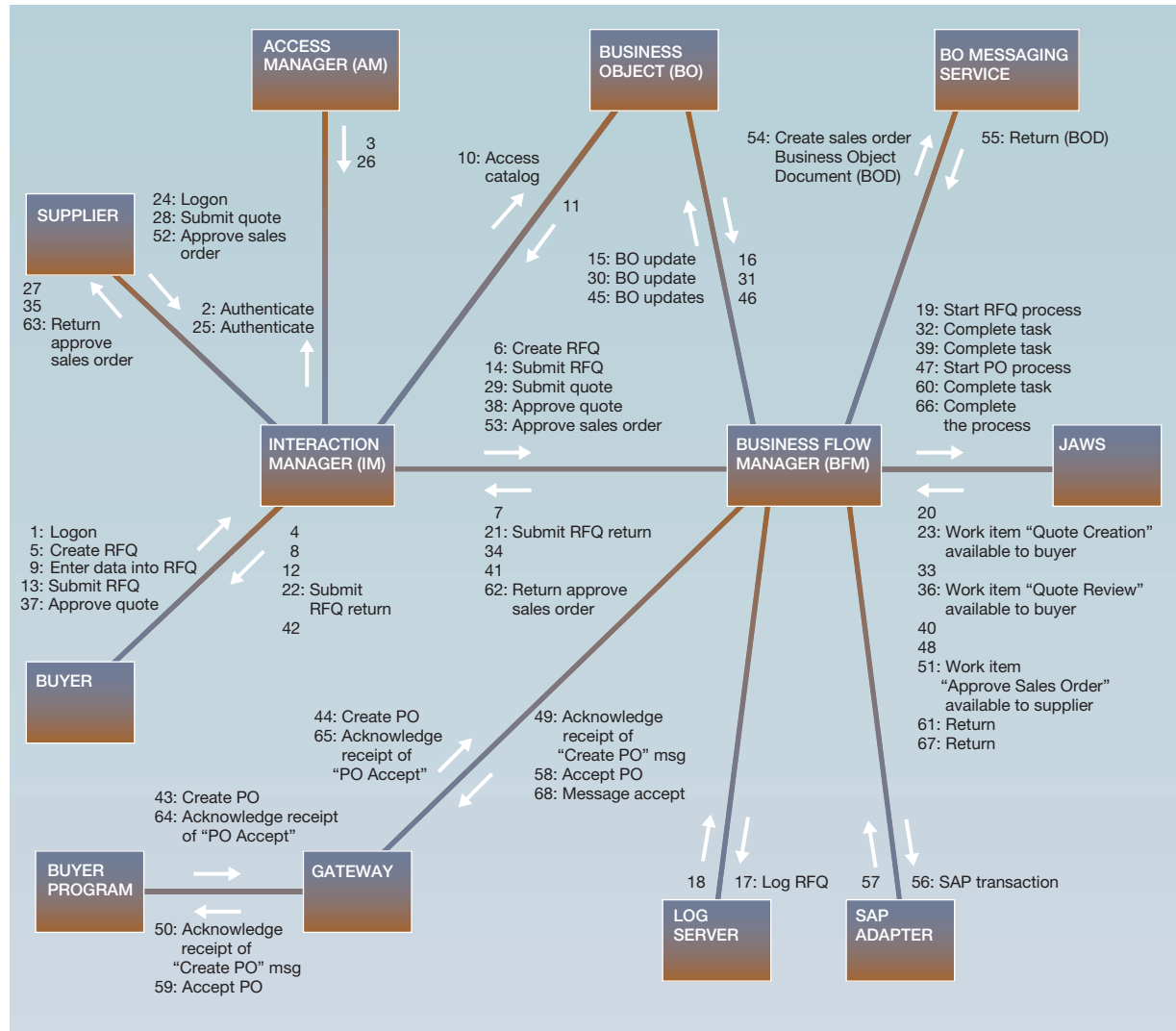
Event- and trigger-driven processing. The realization of an ADoc as a state machine with event-driven state transitions supports an event-driven programming model.

Concluding remarks

The platform proposed in this paper addresses many of the challenges in the development of state-of-the-art PTX solutions. We conclude with a discussion of the outstanding issues, some of which are being addressed by our own ongoing work in this area.

Partner enablement is critical to the success of a private exchange. Unless the trading partners of a company are able to connect and transact using the PTX platform, the business benefits of a PTX solution are not materialized. Typically, the onus of bringing the partners on board rests with the hosting company. Hence, the PTX platform needs to provide the tools that make it easy for trading partners to come on board. This is a hard problem because of the variability in the participants’ capabilities and sophistication in the required technologies. Adding to the complexity is the need for a set of customized features and functions to be presented for each partner. The tools should facilitate the customization of business rules, processes, and workflows for each participant. Our current work attempts to solve this problem by creating a “service provisioning” component in the platform that alleviates some of the difficulties in bringing a new trading partner on

Figure 16 Sequence diagram for a sample PTX RFQ



board. This component uses Web services technology and standards to accelerate partner enablement.

We have briefly mentioned the PTX Solution Builder as the build-time component of our platform. Current implementation of this module is too low-level, requiring J2EE** (Java 2 Platform, Enterprise Edition) development skills to create or customize a solution template. We are working on a high-level meta-model to describe the solution template and associated tools, which can be used without programming skills to develop and customize a solution.

Another area that needs work is solution-centric management of the PTX solution. The solution management component needs to be enhanced to manage the service level agreements (SLAs) that the company has with its trading partners. Auditing information at various levels—system level, transaction level, application level, process level—are needed for effective SLA management. The audit log is an excellent source for business intelligence as well. We are currently working on an agent-based PTX system that exploits workflow audit logs for intelligent trading partner management.

*Trademark or registered trademark of International Business Machines Corporation.

**Trademark or registered trademark of the Object Management Group, Sun Microsystems, Inc., RosettaNet, JAWS Technologies Inc., or SAP AG.

Cited references and notes

1. AMR Research, "The Year of the Private Trading Exchange," *The Report on B2B Marketplaces*, 27–35 (January 2001).
2. IBM WebSphere Business Integrator, available from IBM Corporation; see <http://gwareview.software.ibm.com/software/webserver/btobintegrator/library.html>.
3. B. Zoellink, "eMarkets: Fixing What Is Broken," *Business Analysis*, Fasterwater LLP (2000).
4. *Automotive Private Exchange POC High-Level Requirements and Design*, IBM internal report (2001).
5. *Electronics Private Exchange POC High-Level Requirements and Design*, IBM internal report (2001).
6. *RosettaNet Implementation Framework Specification*, Version 1.1 (November 1999), available from RosettaNet; see <http://www.rosettanet.org>.
7. *Collaborative Planning, Forecasting, and Replenishment Voluntary Guidelines* (June 1998), available from Voluntary Industry Commerce Standards; see <http://www.cpfr.org>.
8. D. Miklovic and R. Altman, "Collaborative Commerce Poses Technological Challenges," *Strategy & Tactics/Trends & Direction*, Gartner Report COM-13-1373 (February 28, 2001).
9. Ross Altman, "Total Business Integration," *Strategy & Tactics/Trends & Direction*, Gartner Report COM-13-0013 (February 2, 2001).
10. D. McCoy and J. Thompson, "Application Integration Thriving Among Standards," *Strategy & Tactics/Trends & Direction*, Gartner Report AV-14-0031 (July 6, 2001).
11. J. Correia, "Looking Forward to 2005 in the AIM Market," *Strategy & Tactics/Trends & Direction*, Gartner Report SOFT-WW-DP-0019 (July 31, 2001).
12. J. Correia, "What Is Happening in the Infrastructure Software Market?" *Strategy & Tactics/Trends & Direction*, Gartner Report AV-14-1931 (August 1, 2001).
13. R. Schulte and R. Altman, "Application Integration: Success Amid Turmoil," *Strategy & Tactics/Trends & Direction*, Gartner Report AV-14-2581 (August 8, 2001).
14. R. Schulte and R. Altman, "Integration Middleware Vendors Mix and Match Functionality," *Strategy & Tactics/Trends & Direction*, Gartner Report QA-14-2662 (August 7, 2001).
15. See <http://www.webmethods.com/>.
16. See <http://www.vitria.com/>.
17. Common Object Request Broker Architecture[®]/Internet Inter-ORB[™] Protocol.
18. See <http://www.tibco.com/>.
19. K. Bhaskaran, J.-Y. Chung, R. Das, T. Heath, S. Kumaran, and P. Nandi, "An e-Business Integration & Collaboration Platform for B2B e-Commerce," *Proceedings, 3rd International Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems*, San Jose, CA (June 21–22, 2001).
20. E. Gamma, R. Helm, R. Johnson, J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley Publishing Co., Reading, MA (1995).
21. R. H. High, "Component Model for Managed Objects in Large-Scale Distributed Systems," *Component-Based Software Engineering*, Thomas Jell, Editor, Cambridge University Press, Cambridge, MA (1998), pp. 117–136.
22. R. E. Johnson, "Components, Frameworks, Patterns," *Proceedings, Symposium on Software Reliability*, Boston, MA (May 17–20, 1997), pp. 10–17.
23. N. Nayak, K. Bhaskaran, and R. Das, "Virtual Enterprises—Building Blocks for Dynamic e-Business," *Proceedings, Workshop on Information Technology for Virtual Enterprises*, Gold Coast, Queensland (February 2, 2001), pp. 80–87.
24. S. Feldman, "Electronic Marketplaces," *IEEE Internet Computing* 4, No. 4, 93–95 (July/August 2000).
25. C. Bornhovd, "Semantic Metadata for the Integration of Web-Based Data for Electronic Commerce," *Proceedings, International Conference on Advance Issues of E-Commerce and Web-Based Information Systems*, Santa Clara, CA (April 8–9, 1999), pp. 137–145.
26. M. Themistocleous and A. Poulymenskou, "Developing Electronic Stores: Some Critical Considerations for Practice," *Proceedings, 9th International Workshop on Database and Expert Systems Applications*, Vienna, Austria (August 26–28, 1998), pp. 667–672.
27. B. F. Schmid and M. A. Lindemann, "Elements of a Reference Model for Electronic Markets," *Proceedings, 31st Hawaii International Conference on System Sciences*, Kohala Coast, HI (January 6–9, 1998), pp. 193–201.
28. D. Gawlick, "Infrastructure for Web-based Application Integration," *Proceedings, 17th International Conference on Data Engineering*, Heidelberg, Germany (April 2–6, 2001), pp. 473–476.
29. R. Peacock, "Enabling Technologies for e-Business," *IT Professional* 2, No. 4, 52–54 (July–August 2000).
30. T. N. Nguyen, "Modeling EAI-Based e-Business Solutions," *Proceedings, IEEE International Conference on Systems, Man, and Cybernetics*, Nashville, TN (October 8–11, 2000), pp. 286–291.
31. S.-M. Huang, I. Kwan, D. C. Yen, and S.-Y. Hsueh, "Developing an XML Gateway for Business-to-Business Commerce," *Proceedings, 1st International Conference on Web Information Systems Engineering*, Hong Kong (June 19–20, 2000), pp. 67–74.
32. Java Message Service API; see <http://java.sun.com/products/jms/>.
33. Java Authentication and Authorization Service, Version 1.0, see <http://java.sun.com/security/jaas/>.
34. G. Flurry and W. Vicknair, "The IBM Application Framework for e-Business," *IBM Systems Journal* 40, No. 1, 8–24 (2001).
35. WebSphere application server, IBM Corporation; see <http://www.ibm.com/software/webserver/>.
36. WebSphere portal server, IBM Corporation; see <http://www.ibm.com/software/webserver/portal/>.
37. WebSphere MQSeries family, IBM Corporation; see <http://www.ibm.com/software/ts/mqseries/>.
38. Tivoli products, see <http://www.tivoli.com/products/>.
39. Enterprise JavaBeans technology, Sun Microsystems, Inc.; see <http://java.sun.com/products/ejb/>.

Accepted for publication January 16, 2002.

Santhosh Kumaran IBM Research Division, Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, New York 10598 (electronic mail: sbk@us.ibm.com). Dr. Kumaran received the B.Tech. degree in naval architecture from the University of Cochin, the M.S. degree in ocean engineering from the University of Miami, and the Ph.D. degree in computer science from Oregon State University. He currently manages the e-commerce systems research team at IBM's T. J. Watson Research Center in Yorktown. He has been with IBM since 1996 and was awarded

an IBM Outstanding Technical Achievement award in 2001 and an Execute Now award in 2002. His research interests include frameworks and platforms for e-business solutions, business process tooling, performance, scalability, and availability analysis of complex e-business systems, and dynamic e-business models.

Ying Huang *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, New York 10598 (electronic mail: ying_huang@us.ibm.com).* Dr. Huang received the B.S. degree in applied mathematics from Fudan University and the Ph.D. degree in operations research from Stanford University. He currently is a research staff member at the T. J. Watson Research Center. He has worked previously with other IBM divisions and with Philips Research. He has conducted research, development, and customer engagements in electronic commerce and marketplaces, supply chain modeling, and decision support systems. Areas of current research include integration architectures, e-marketplaces, and analysis and mark-up languages for business-to-business data and decision exchanges.

Jen-Yao Chung *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, New York 10598 (electronic mail: jychung@us.ibm.com).* Dr. Chung received the B.S. degree in computer science and information engineering from National Taiwan University and the M.S. and Ph.D. degrees in computer science from the University of Illinois at Urbana-Champaign. Since June 1989, he has been with the T. J. Watson Research Center as a research staff member. He currently is the senior manager of the electronic commerce and supply chain department, and program director for the IBM Institute for Advanced Commerce Technology office. He has authored or coauthored over 60 technical papers in published journals or conference proceedings. He has been involved in research, development, and customer engagements in electronic commerce, electronic marketplaces, and Web application systems. Dr. Chung's current research is in e-marketplaces, XML and EDI, integrating existing business with the World Wide Web, back-end ERP (enterprise resource planning) integration, Web data access, and scalable Web servers. He was awarded an IEEE Outstanding Paper award in 1995, two IBM Outstanding Technical Achievement awards, in 1994 and 2000, an IBM Outstanding Contribution award in 1997, and three IBM Research Division awards, in 1990, 1996, and 2001. He is a senior member of the IEEE and a member of ACM.